MATH/STAT 4450/8456 Machine Learning Competition #2

Predicting the fraud in self-checkout stores

The second contest for this course is to predict the fraud in self-checkout stores. In self-checkout stores or smart stores (like Amazon Go), customers can enjoy their shopping experience without long lines at the checkout counter. They can even use their own smart phone to scan the products. The scenario for our competition is a grocery store that allows customers to scan their products using a hand-held mobile scanner while shopping. However, some customers do not scan all of the items in their cart intentionally or inadvertently. Our goal is to detect those fraudulent purchases.

Kaggle website

The data can be downloaded from Canvas and the Kaggle site and you will have to participate the competition through the link (https://www.kaggle.com/t/81c95ee8e0d749f4b2a7a710864bf85a) as it is not open to public.

A few things:

- Please use your real name as the "team" name when making submissions, as I need to know who you are to get your grade. You will work individually so you are the only member of your team.
- The maximum daily submission number is 20. So you will need to wait until the next UTC day after submitting 20 results.
- There is a public leaderboard (10% of the test set) and a private leaderboard (90% of the test set). Your final grade will be based on the private leaderboard score.

Description of variables

- id: row ID. Has no meaning.
- credit: Customer's credit level. Either low or high.
- duration: The time in seconds between the first and last scans.
- total: The total amount of all scanned and not cancelled products.
- scans: The number of all scanned and not cancelled products.
- voidedScans: The number of voided scans.
- attemptsWoScan: The number of attempts to activate the scanner without scanning any products.
- modifiedQuantities: The number of modified quantities for the scanned products.
- fraud: Whether the purchase is fraudulent or not. 1: fraud. 0: not fraud.

Data

Here is a quick look at the data.

```r
train = read.csv("data/train.csv")
str(train)
```

```
## ’data.frame':    8305 obs. of  9 variables:
##  $ id                : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ credit            : chr  "High" "High" "High" "High" ...
##  $ duration          : int  273 1481 278 576 356 331 1708 271 1777 353 ...
##  $ total             : num  82.1 57.8 50.1 85.3 58.8 ...
##  $ scans             : int  5 21 10 16 10 22 22 24 1 9 ...
##  $ voidedScans       : int  11 10 1 10 4 3 2 4 6 4 ...
##  $ attemptsWoScan    : int  9 4 1 9 3 7 5 4 4 6 ...
##  $ modifiedQuantities: int  0 1 5 4 2 0 0 2 3 1 ...
##  $ fraud             : int  0 0 0 0 0 0 0 0 0 0 ...
```

```r
table(train$fraud)
```

```
##
##    0    1
## 7143 1162
```

The number of non-fraud cases is about 6 times the number of fraud cases. So this is an unbalanced classification problem. Even if you predict all cases by 0, you will obtain around 85% correct rows. Therefore we will use the weighted accuracy to evaluate your result. You may consider techniques like upsampling, subsampling, class weights, regularization, etc., to deal with the unbalanced problem.

```r
test = read.csv("data/test.csv")
str(test)
```

```
## ’data.frame':    158000 obs. of  8 variables:
##  $ id                : int  8306 8307 8308 8309 8310 8311 8312 8313 8314 8315 ...
##  $ credit            : chr  "High" "Low" "Low" "High" ...
##  $ duration          : int  891 352 712 1681 373 268 231 623 788 1580 ...
##  $ total             : num  18.1 10.6 84.2 24.9 10.4 ...
##  $ scans             : int  9 5 5 11 6 5 24 16 14 4 ...
##  $ voidedScans       : int  11 0 9 4 8 2 2 8 2 10 ...
##  $ attemptsWoScan    : int  10 9 6 6 1 10 1 6 5 8 ...
##  $ modifiedQuantities: int  0 0 1 4 0 0 4 4 1 1 ...
```

The test set is about 9 times the size of the training set. About 1/9 of the test set is used in the public leaderboard.

Evaluation metrics

Instead of using the accuracy for evaluation, we use weighted accuracy given by the formula below.

$$\text{Weighted accuracy} = \sum_{i=1}^{N} w_i I(y_i = \hat{y}_i)$$

where

$$w_i = \begin{cases} 1 & \text{real class is non-fraud} \\ 6 & \text{real class is fraud} \end{cases}$$

Task

1. Create the most accurate classifier that you can for the data, as measured by the test data.
2. Write a **10-18 page** slides summarizing your approach to
   (a) exploring and preparing the data for modeling,
   (b) formulating the model design matrix (input matrix / predictor space),
   (c) building the model and tuning parameters,
   (d) validating the model by training & validation sets, or other approaches,
   (e) comparing results from all attempts,
   (f) findings from the data and challenges from this contest.

Kaggle submission

Your Kaggle submission file should be in the csv format with two columns: `id` and `fraud`. Example of the submission:

```
id,fraud
8306,0
8307,0
8308,1
...
166305,0
```

Canvas submission

1. Code. The code you used for the BEST Kaggle submission (not for all submissions). Should be `.R`, `.Rmd`, `.py`, or `.ipynb` files. Make sure you include detailed explanation of the steps.
2. Slides for the presentation. Can be a `.pptx`, `.pdf`, or `.html` file.

Deadlines

- May 13 (11:59 pm): Final prediction submission on Kaggle.
- May 14 (5 - 6 pm): Presentations. Only the top 6 participants will present the result. Each presentation should be around 8 minutes. Winners will be announced at the presentation time.
- May 15 (11:59 pm): Slides and code submission.

Grading:

- Total points: 20
  - Accuracy of classifier: 8
    - Score will be curved based on your highest accuracy.
  - Progress made from multiple submissions: 3
    - Number of good submissions (highest accuracy in all previous submissions): 2
    - Amount of decreasing of the error rate: 1
  - Presentation: 8
    - Data exploration and visualization: 1
    - Data preparation and feature engineering: 1
    - Design matrix: 1

    - ∗ Validation approach: 1
    - ∗ Model selection: 1
    - ∗ Parameter tuning: 1
    - ∗ Result table for all models & parameters: 1
    - ∗ Summary and interesting findings: 1
- Met the deadlines: 1